THE JOURNAL OF PHYSICAL CHEMISTRY A
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/JPCA

Article

Multiscale Physics-Informed Neural Networks for Stiff Chemical Kinetics

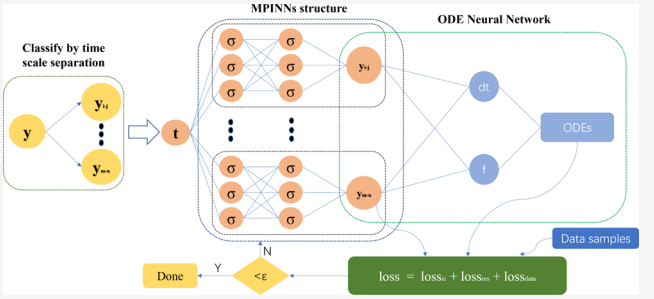
Yuting Weng and Dezhi Zhou*

Cite This: J. Phys. Chem. A 2022, 126, 8534−8543

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: In this paper, a multiscale physics-informed neural network (MPINN) approach is proposed based on the regular physics-informed neural network (PINN) for solving stiff chemical kinetic problems with governing equations of stiff ordinary differential equations (ODEs). In MPINNs, chemical species with different time scales are grouped and trained by multiple corresponding neural networks with the same structure. The adaptive weight based on a key performance indicator is assigned to each loss term when calculating the summation of loss residues. With this structure, MPINNs provide a framework to solve challenging stiff chemical kinetic problems without any stiffness-removal artifacts before training. In addition, by introducing a small number of ground truth data (GTD) points (less than 10% of the number required for residual loss calculation) and adding data loss terms into loss functions, MPINNs show superior ability to represent stiff ODE solutions at any desired time. The accuracy of MPINNs is tested with classical chemical kinetic problems, by comparing with the regular PINN and other state-of-the-art methods with special consideration for solving stiff chemical kinetic problems with PINNs. The validation results show that MPINNs can effectively avoid the influence of stiffness on neural network optimization. Compared with the traditional deep neural network only trained by GTD, MPINNs can use no data or a relatively small amount of data to achieve high-precision prediction of stiff chemical ODEs. The proposed approach is very promising for solving stiff chemical kinetics, opening up possibilities of MPINN application in different fields involving stiff chemical dynamics.

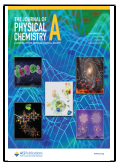


INTRODUCTION

Chemical kinetics are very important in different fields such as chemical engineering, life sciences, and environmental engineering. Solutions of the governing equations for chemical kinetic problems require numerical ordinary differential equation (ODE) solvers, leading to expensive computation. Therefore, deep learning has been widely used in recent years to solve ODEs and to predict concentrations of chemical species in chemical kinetic problems.[1−3] Specifically, considering the large amount of data involved in the process of solving the typically high-dimensional chemical kinetic problems, the advantage of the deep learning method prevails compared with conventional solvers.[4] Therefore, some deep learning methods for chemical kinetic problems can be seen in the recent literature. For example, Brown et al.[5] proposed a new deep neural network (DNN) structure to solve chemical ODEs; Zhang et al.[6] used the DNN based on a multiscale sampling method to solve chemical kinetic problems; Dikeman et al.[7] used a combination of stiffness-reduced autoencoder and neural ODE models to solve combustion chemical kinetics problems.

However, the success of traditional deep learning neural network approaches on solving ODEs significantly relies on the quality and size of data. In the two methods referred to in the previous paragraph, Brown et al.[5] and Zhang et al.[6] have to propose complex data sampling methods to ensure training data quality with the basis of sufficient data points. In addition, training for different chemical kinetic problems requires the generation of a large number of data repeatedly, which makes it difficult to ensure the comprehensiveness and robustness of data under different conditions. Therefore, the physics-informed neural network (PINN) has attracted extensive attention recently.[8] The feature that the PINN does not rely on a large amount of data mainly stems from the fact that the residual of the physical governing equations (usually differential equations) is taken as the loss function instead of ordinary data loss. This method gets rid of the dependence on a large number of data and is able to solve differential equations (i.e., forward problems) and inferring parameters based on observations (i.e., inverse problems). The PINN has

Received: September 12, 2022
Revised: October 20, 2022
Published: November 2, 2022



a good application prospect in the fields of fluid,[9] wave field,[10] and materials.[11] Raissi et al.[12] first proposed the PINN to solve forward and inverse problems of partial differential equations. Li and Mei[13] added a data loss item to PINN loss function and pointed out that the accuracy of the neural network can be improved through small sample learning combined with the PINN. In this structure, the number of data samples as the loss item is much smaller than that required for calculating residual loss. Some studies have also combined the PINN with other learning methods. For example, Chakraborty[14] combined the PINN with transfer learning and proposed to train the PINN by data with different fidelities, that is, to train the whole network with low-fidelity data first and then fine-tune the network with high-fidelity data.

However, there are still some major concerns to be addressed in the application of the PINN for stiff ODEs. Wang et al.[15] analyzed the basic failure mode of the PINN, pointed out that the failure mode was related to the numerical stiffness of the back-propagation of the gradient that caused imbalance during model training, and proposed an annealing algorithm based on learning rate to balance the interaction between different loss terms. Xiang et al.[16] proposed an adaptive weight modification method on loss function which can avoid the imbalance during the training process of the PINN. The difficulty of applying the PINN to solve chemical kinetic problems also comes from the stiffness caused by the large time scale span of different chemical species. To tackle this problem, Ji et al.[17] proposed to apply quasi-steady-state assumption (QSSA) before the training process of the PINN to reduce the stiff chemical species. This method needs to use non-trivial artifacts (i.e., QSSA) to remove stiff chemical species in advance, limiting the application of this method to small stiff chemical kinetic problems. Very recently, an X-TFC framework[18] has been proposed to solve stiff chemical kinetics problems without artifacts. This framework expands the control functions and hardcodes the initial condition into the control function. It uses an extreme learning machine (ELM) to predict the results. Although the X-TFC framework improves the prediction accuracy for stiff chemical kinetics problems, it requires the combination of the PINN with theory of functional connections and ELM, leading to non-trivial transformation of ODEs to constrained expressions (especially for large-size chemical kinetic problems).

In this work, a multiscale PINN (MPINN) approach is proposed, aiming to learn the solutions of chemical kinetic problems governed by stiff ODEs using redesigned but simple PINN structures. The chemical species of different time scales are classified, and the loss function is constructed separately. This framework uses different neural networks with the same structure to fit chemical species in different groups, so as to avoid the influence of stiffness in the training process of neural networks. Adaptive weight summation based on a key performance indicator for residual losses is proposed to balance the decline of loss. In addition, we show that a supplement of small ground truth data (GTD) samples to the loss function can further improve the effectiveness of network training. The MPINN approach is verified and analyzed on the ROBER problem, Chemical Akzo Nobel problem, and POLLU problem, showing high prediction accuracy in solving stiff chemical kinetic problems.

The paper is organized as follows. In the Methodology section, the general MPINN methodology is presented, explaining the MPINN structure, adaptive weighted summation of losses, and the introduction of GTD loss. Accuracy of the proposed MPINNs is then tested on several stiff chemical kinetic problems against benchmarks from literature studies in the Results and Discussion section. Finally, conclusions and outlook of future work are presented in the Conclusions section.

METHODOLOGY

Physics-Informed Neural Network. Raissi et al.[12] proposed the PINN to solve differential equations. In this paper, we apply the PINN to the chemical kinetic system. For a homogeneous chemical reaction system, the following ODEs are the general form of governing equations

$$\mathrm{d}\boldsymbol{y}/\mathrm{d}t = f(t, \boldsymbol{y}), \qquad t_0 \le t \le t_{\text{final}} \tag{1}$$

$$\boldsymbol{y}(t_0) = \boldsymbol{y}_0 \tag{2}$$

where $\mathbf{y} = [y_1, y_2, ..., y_N]^T$ is the column vector of chemical species concentration and N is the number of chemical species. Equation 2 shows the initial condition of the reaction system. According to the idea of Raissi et al.,[12] $\boldsymbol{y}$ can be approximated by a DNN $f_\theta(t)$, and the residual loss of eq 1 is defined as

$$\text{loss}_{\text{res}} = \frac{\mathrm{d}}{\mathrm{d}t} f_\theta(t) - f(t, \boldsymbol{y}) \tag{3}$$

The derivative of the output $f_\theta(t)$ of the neural network with respect to t can be obtained by automatic differentiation. Then, the loss function of the whole neural network is defined as

$$\text{loss} = \text{loss}_{\text{initial}} + \text{loss}_{\text{res}} \tag{4}$$

$$\text{loss}_{\text{initial}} = \sum_{i=1}^{N} [f_{\theta,i}(t = t_0) - y_i(t = t_0)]^2 \tag{5}$$

The residual loss will be calculated by randomly sampling a certain number of time points in $[t_0, t_{\text{final}}]$, and the initial condition loss will be calculated by sampling a certain number of time points where $t = t_0$.

Multiscale PINN Structure. Considering the stiffness of some chemical kinetic systems, a single PINN cannot well fit the concentrations of various chemical species with different time scales. Without losing generality, we use the classical chemical reaction kinetic problem—ROBER problem as an example to illustrate numerical stiffness in chemical kinetics. The ROBER problem can be described by the following reaction equations

$$\mathrm{A} \xrightarrow{k_1} \mathrm{B},\ \mathrm{B} + \mathrm{B} \xrightarrow{k_2} \mathrm{C} + \mathrm{B},$$
$$\mathrm{B} + \mathrm{C} \xrightarrow{k_3} \mathrm{A} + \mathrm{C} \tag{6}$$

where $k_1 = 0.04$, $k_2 = 3 \times 10^7$ and $k_3 = 10^4$. We can easily find that $k_2/k_1 \sim 10^9$, that is, the reaction rate constant changes within the range of 9 orders of magnitude, resulting in the huge stiffness of the entire chemical reaction system. Using the PINN to predict the concentrations of different species in the ROBER problem is difficult since the huge stiffness causes the corresponding loss functions of different chemical species to be in different orders of magnitude, which leads to the competition between learning different chemical species. More specifically, during the optimization process of the neural networks, unbalanced loss functions of different

https://doi.org/10.1021/acs.jpca.2c06513
J. Phys. Chem. A 2022, 126, 8534−8543

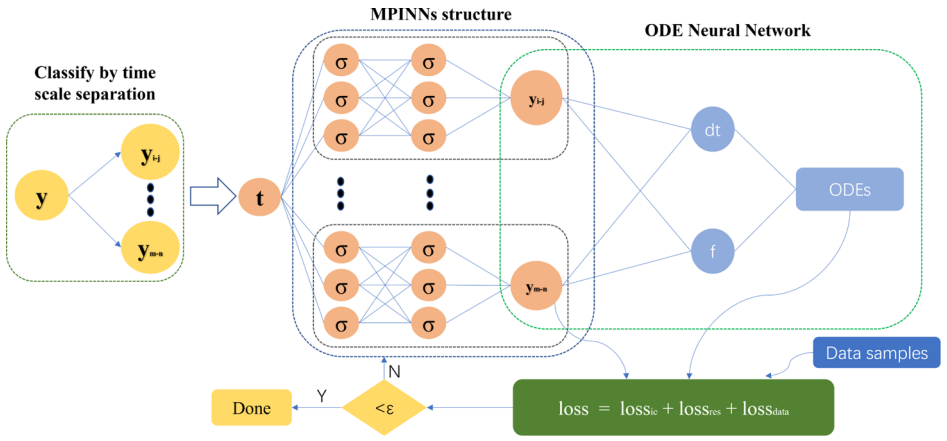


Figure 1. Schematic diagram of the structure of MPINNs.

chemical species will result in unbalanced gradient backward and parameter update.[15]

Therefore, this paper proposes to use multiple neural network models with the same structure to fit the concentration changes in chemical species with different magnitude orders of time scale. All chemical species are divided into different categories based on the difference of their time scale orders of magnitude, and each category has its own neural network structure. The time scale w of each chemical species will be calculated as

$$w_i = \frac{y_{\max^i}}{\mathrm{d}y_i/\mathrm{d}t} \tag{7}$$

where i means the i-th chemical species in the chemical kinetic system and $y_{\max}$ means the maximum concentration of the chemical species in the solution time range. The number of categories will be determined based on the number of species time scale orders of magnitude. The loss function of each neural network is the residual loss and boundary loss of the chemical species in the corresponding category. Therefore, we have

$$\begin{aligned} \mathrm{loss}_{\mathrm{res}}^{k} &= \sum_{i\in\Omega} \mathrm{loss}_{\mathrm{res}}^{i} = \sum_{i\in\Omega} \left[\frac{\mathrm{d}}{\mathrm{d}t} f_{\theta,i} - f(t, \mathbf{y})\right]^2, \\ \mathrm{loss}_{\mathrm{initial}}^{k} &= \sum_{i\in\Omega} \mathrm{loss}_{\mathrm{initial}}^{i} = \sum_{i\in\Omega} [f_{\theta,i}(t = t_0) - y_i(t = t_0)]^2 \end{aligned} \tag{8}$$

where k represents the k-th category of all chemical species and Ω represents the collection of chemical species in the k-th category. Note that when evaluating the residual loss for the k-th category, the information from other categories (see $f(t,\mathbf{y})$ in eq 8) is needed. In this sense, the segregation of species with different time scales does not hinder the cross-talk among groups and thus not harm the physics information during the NN training in MPINNs.

The sum of losses is then minimized, in which the residual losses of different neural networks are added with adaptive weights to balance the impact of loss terms in different categories. This weight will be adaptively modified in the iterative optimization process. In detail, all loss terms will be normalized to the same order of magnitude by the maximum loss value, and a weight term is given to the residual loss of each category as follows[19]

$$F_k(p_c; \gamma_0) = -(1 - p_c)^{\gamma_0} \log(p_c) \tag{9}$$

where $F_k(p_c; \gamma_0)$ is the weight of residual loss for the k-th category. p_c is the key performance indicator, and it can be obtained from the GTD sample loss after introducing the data sample loss (see the next section). If GTD sample loss is not introduced, we can use the decline rate of the residue loss term in each category to measure p_c.[20] The magnitude of p_c is negatively correlated with the neural network optimization difficulty in the corresponding category. γ_0 is a hyperparameter, and the value of γ_0 is to incline the optimization target to the difficult target as much as possible. Following the setting by Guo et al.,[19] its value is set to 0.1 in this paper. Finally, the total residual loss for all the K categories of MPINNs can be written as

$$\mathrm{loss}_{\mathrm{res}} = \sum_{k=1}^{K} F_k \times \mathrm{loss}_{\mathrm{res}}^{k} \tag{10}$$

Introduction of GTD Samples. In order to achieve more accurate prediction, we introduce a small number of GTD samples into the training process and thus take the mean square error between the network output and GTD as an additional term of the loss function for each category (i.e., $\mathrm{loss}_{\mathrm{data}}^{k}$). With this additional data loss term, we can obtain the value of p_c by the ratio of data loss to the prediction results in each category. In this way, by summing up the loss functions in each category, the total loss function becomes

$$\mathrm{loss}_{\mathrm{total}} = \sum_{k=1}^{K} \mathrm{loss}_{\mathrm{initial}}^{k} + \sum_{k=1}^{K} \mathrm{loss}_{\mathrm{data}}^{k} + \mathrm{loss}_{\mathrm{res}} \tag{11}$$

where K is the number of categories in the stiff chemical kinetic problem. The loss function composed of data loss provides the setting of prior information on the neural network solution. More specifically, it restricts the space of acceptable solutions to a manageable size as a regularization term. With this method, the known model can be rectified and the generalization ability can be improved in the scenario of a small number of GTD points provided. Compared with neural networks only trained by GTD loss (e.g., traditional DNN), MPINNs require smaller size of data and significantly improve accuracy of the model. For future application of MPINNs in real stiff systems where the species information (e.g., $y_{\max^i}$) is unknown, the GTD points can also play a role in evaluating the species time scales as shown in eq 7. It is also noted that the

https://doi.org/10.1021/acs.jpca.2c06513
J. Phys. Chem. A 2022, 126, 8534−8543

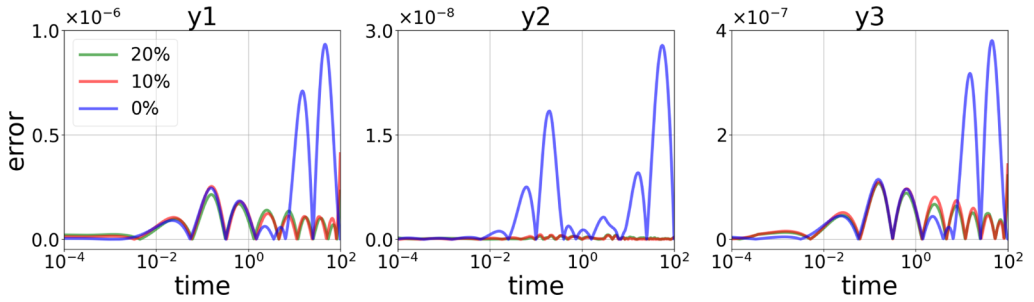


Figure 2. Mean squared error of MPINNs with GTD point introduction of 0, 10, and 20% of the number required for residual loss calculation, compared with the true value obtained by CVODE in the ROBER problem.

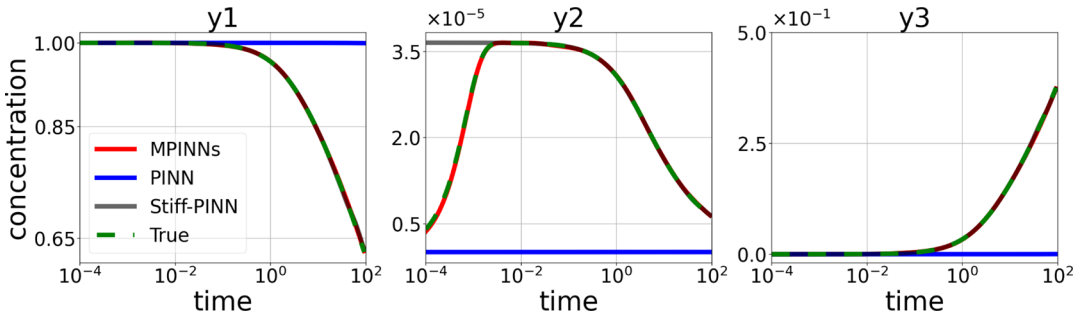


Figure 3. Prediction results of three chemical species concentrations in the ROBER problem obtained by PINNs, Stiff-PINNs, and MPINNs, where the true value (true) is obtained by CVODE.

inclusion of GTD points would make the current MPINN approach a hybrid model of the DNN and PINN. However, it will be shown in the later verifications that only a very small number of GTD points without any special data sampling method are needed to ensure the great accuracy of MPINNs, eliminating the strong dependence on data size and special data sampling.

With the introduction of GTD samples, the final structure of MPINNs proposed in this paper is shown in Figure 1.

The MPINNs' structure, shown in Figure 1, is a multiscale neural network model, which takes only time (t) as the input. Different from the regular PINN, it uses multiple neural networks to predict chemical species with different time scales and calculate the loss on initial conditions. Combined with the ODE neural network, the residual loss of the governing equation is calculated, and the data loss based on a small number of GTD samples is added to obtain the final loss function. In the optimization process of the loss function, termination of the training process is determined by the preset maximum iteration times, iteration tolerance times, and loss boundary. When iteration times are greater than the maximum iteration times, the loss value is lower than the preset loss boundary or loss begins to increase and still increases after exceeding the iteration tolerance times, and the training process of the neural network is terminated.

Having introduced the formulation of MPINNs, it is worthwhile to have a discussion on the difference of MPINNs and Stiff-PINNs.[17] First, the original motivation of MPINNs is to remedy the issues of unbalanced training of neural networks of PINNs for stiff systems, by redesigning NN structures based on the time scales of species. As for Stiff-PINNs, this pioneering work of the PINN for stiff chemical kinetics is directly trying to remove the stiffness by QSSA, while the original PINN structure is still used for the remaining system without stiffness. Second, in MPINNs, we introduced GTD points and proposed adaptive weights based on a performance indicator. With the aid of these developments, we aim to develop the MPINN for achieving greater prediction accuracy, even without any artifacts before training. Due to these features of MPINNs, it is thus very promising to apply it into real stiff systems (e.g., hydrocarbon combustion), while for Stiff-PINNs, the QSSA procedure could be non-trivial for large stiff ODE systems.

RESULTS AND DISCUSSION

In this section, three classical stiff chemical kinetics problems (ROBER, Chemical Akzo Nobel, and POLLU) are solved by MPINNs, compared with the benchmark cases by the traditional backward difference formula ODE solver (CVODE[21]), and Stiff-PINNs.[17] The comparison of prediction accuracy among MPINNs, PINNs, Stiff-PINNs, and conventional DNNs is shown.

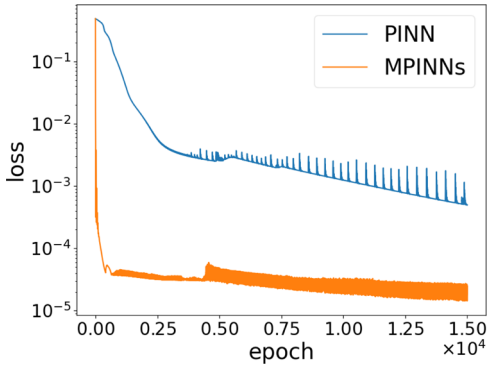


Figure 4. Loss function history of the regular PINN and MPINNs for the ROBER problem.

https://doi.org/10.1021/acs.jpca.2c06513
J. Phys. Chem. A 2022, 126, 8534−8543

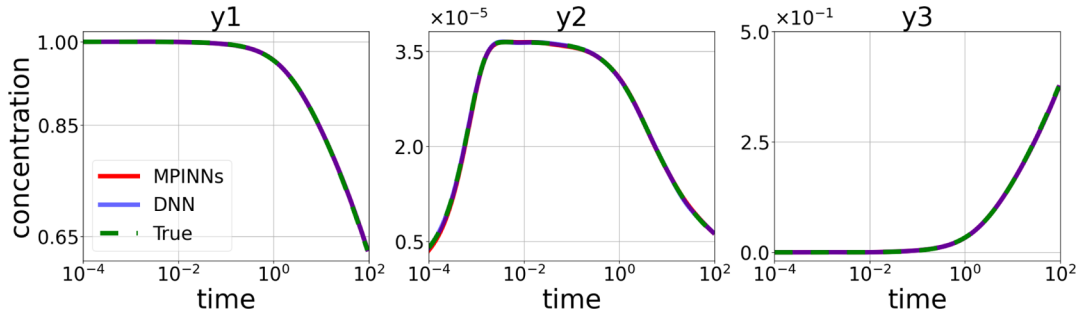


Figure 5. Prediction results of the three chemical species concentrations in the ROBER problem obtained by MPINNs and DNNs, where the true value (true) is obtained by the traditional CVODE solver.

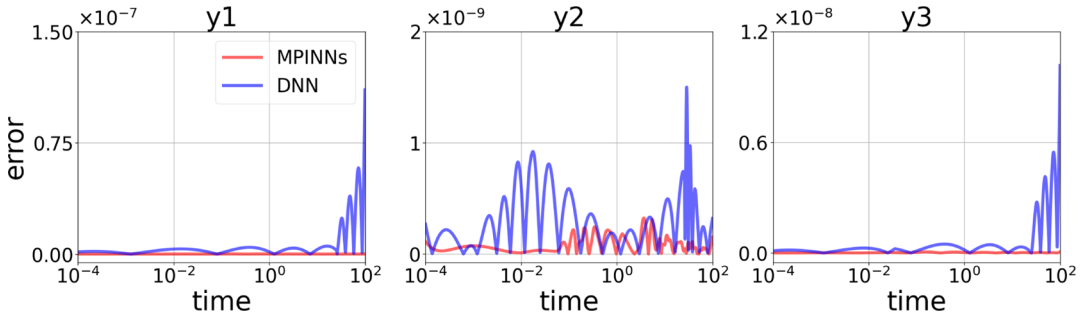


Figure 6. Mean squared error (see eq 16) of the prediction results by MPINNs and DNNs in the ROBER problem.

Table 1. Arithmetic Average Mean Squared Error of Predictions by MPINNs with Different Neural Network Sizes

neurons × layers	arithmetic average mean squared error
32 × 3	2.0475×10^{-6}
64 × 3	8.6374×10^{-6}
32 × 4	7.9651×10^{-6}
64 × 4	1.4583×10^{-5}

Table 2. Time Scale Order of Magnitude for All Chemical Species in the Chemical Akzo Nobel Problem

category	species	time scale
one	y_1, y_3, y_5, y_6	10^1
two	y_2, y_4	10^0

ROBER Problem. For the ROBER problem, the change in chemical species' concentration can be described by the following ODEs

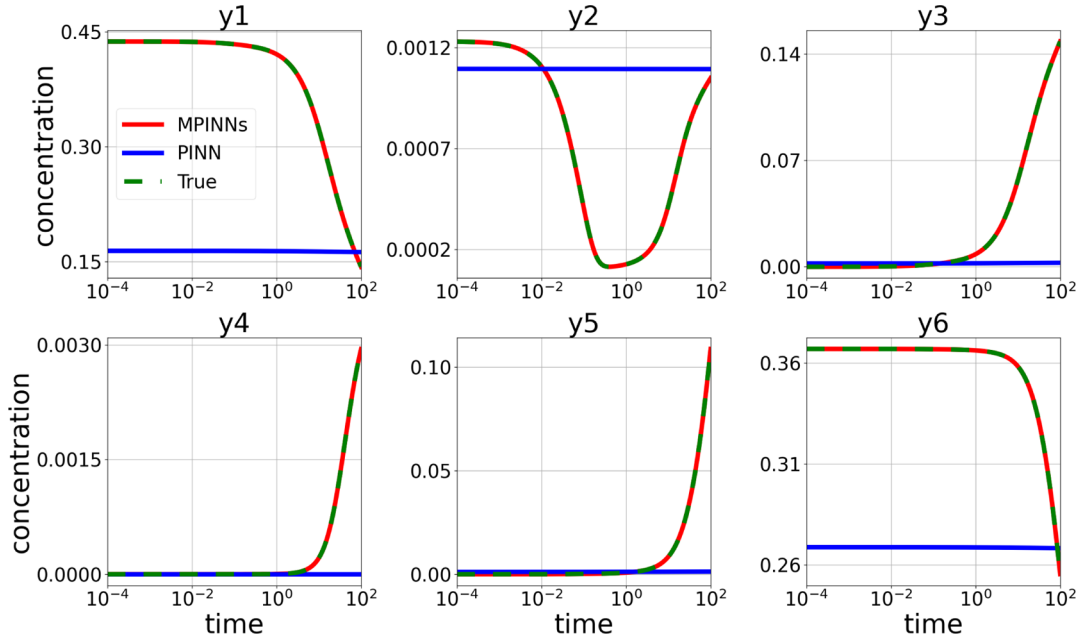


Figure 7. Prediction results of all the six chemical species concentrations in the Chemical Akzo Nobel problem obtained by PINNs and MPINNs, where the true value (true) is obtained by the traditional CVODE solver.

https://doi.org/10.1021/acs.jpca.2c06513
J. Phys. Chem. A 2022, 126, 8534−8543

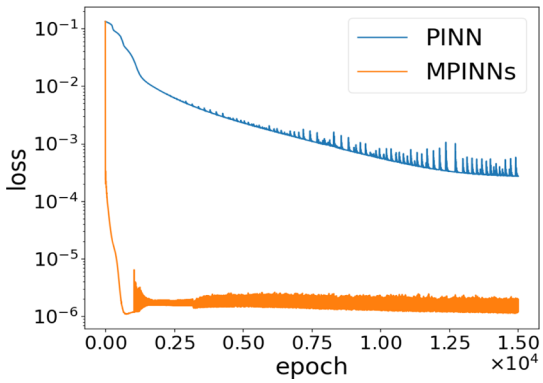


Figure 8. Loss function history of the regular PINNs and MPINNs for the Chemical Akzo Nobel problem.

$$\frac{dy_1}{dt} = -k_1 y_1 + k_3 y_2 y_3,$$
$$\frac{dy_2}{dt} = k_1 y_1 - k_2 y_2^2 - k_3 y_2 y_3, \quad \frac{dy_3}{dt} = k_2 y_2^2 \tag{12}$$

We first build a regular PINN to solve this problem. The neural network accepts time t as the input and outputs the concentrations of three chemical species $[y_1, y_2, y_3]$. In order to avoid imbalance between initial condition loss and the residual loss, we hardcoded the initial conditions into the structure of the neural network following the method in Stiff-PINNs.[17] Therefore, the expression between the outputs and input in the neural network can be written as follows

$$\boldsymbol{y} = \boldsymbol{y_0} + t \times \mathbf{PINN}(t^*) \tag{13}$$

Before training the neural network, it is necessary to scale input t of the neural network. Otherwise, the large magnitude difference of the input variables will cause great interference to the connection weight determination and thus cause the numerical problem when updating parameters. For time t, we calculate its average μ_t and standard deviation σ_t. For each time t_i, we perform the following scaling

$$t_i = \frac{t_i - \mu_t}{\sigma_t} \tag{14}$$

For the centralized and standardized data, we then perform the following processing to achieve input t^* between $[0,1]$

$$t_i^* = \frac{t_i - \min_j t_j}{\max_j t_j - \min_j t_j} \tag{15}$$

Considering the exponential distribution of time, the logarithmic scaling method is also a strategy that can be adopted.

In the training process, only the residual loss of the governing equation and data loss need to be optimized since the initial condition has been hardcoded into the structure of the neural network. The neural network consists of 3 layers, each of which has 32 neurons. The sensitivity of MPINNs' prediction accuracy to the neural network architecture (i.e., number of layers and neurons) is discussed later in this section. Gelu function[22] is adopted as the activation function, and the weight of the neural network is Xaiver-initialized and optimized by the Adam algorithm.[23] A total of 2500 points are randomly sampled within the preset time range as training data.

Based on the proposed MPINN structure and the adaptive weight method for loss function optimization, we first show the sensitivity of MPINNs' prediction accuracy to the number size of ground truth points. Mean squared error between the prediction results and true value is used as the measure target, which will be calculated using

$$\text{error}^i = \frac{\sum_{j=1}^{M} (y^i_{\text{predict},j} - y^i_{\text{true},j})^2}{M} \tag{16}$$

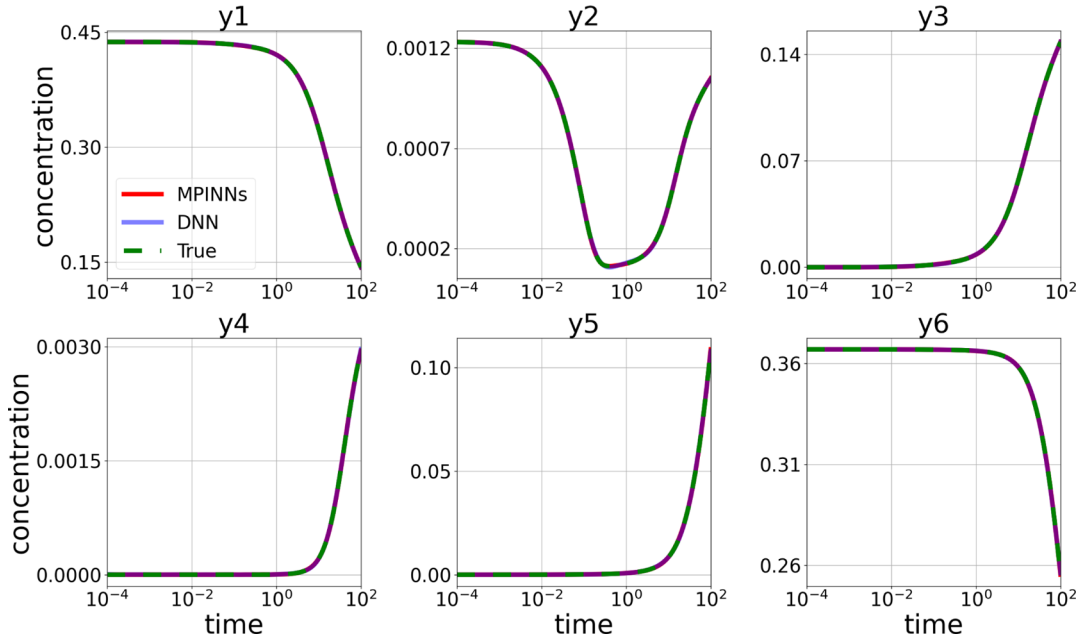


Figure 9. Prediction results of all the six chemical species concentrations in the Chemical Akzo Nobel problem obtained by MPINNs and DNNs, where the true value (true) is obtained by the traditional CVODE solver.

https://doi.org/10.1021/acs.jpca.2c06513
J. Phys. Chem. A 2022, 126, 8534−8543

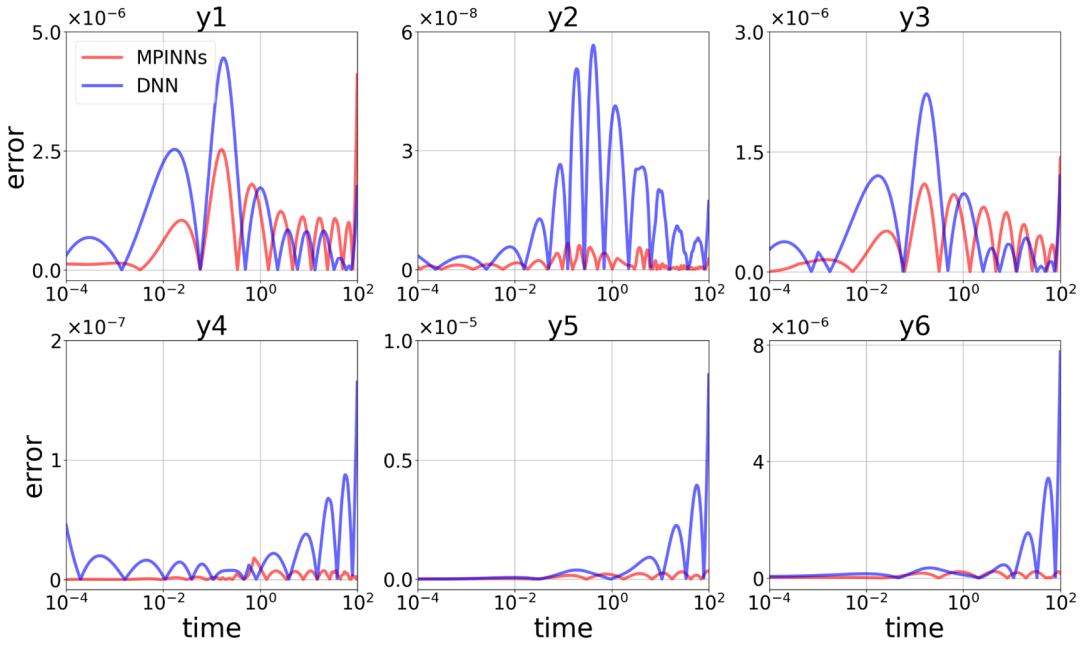


Figure 10. Mean squared error (defined by eq 16) of the prediction results by MPINNs and DNNs in the Chemical Akzo Nobel problem.

Table 3. Time Scale Order of Magnitude for All Chemical Species in the POLLU Problem

category	species	time scale
one	y_3, y_5, y_6, y_{11}, y_{16}	10^{-2}
two	y_{10}, y_{13}, y_{14}, y_{19}, y_{20}	10^{0}
three	y_1, y_2, y_4, y_{12}, y_{18}	10^{1}
four	y_7, y_8, y_9, y_{15}, y_{17}	10^{2}

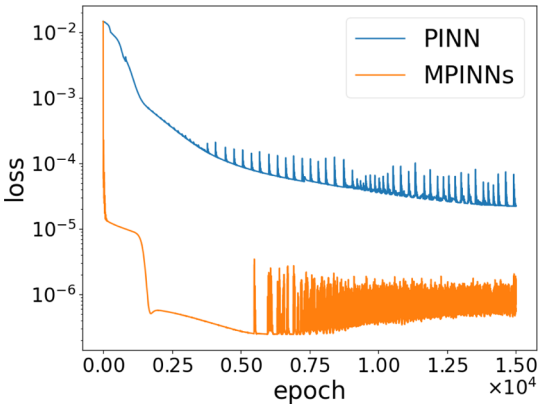


Figure 11. Loss function history of the regular PINNs and MPINNs for the POLLU problem.

where i means the i-th chemical species and j means the j-th data point. M is the total number of data points. In this sense, MPINN training without introducing GTD points and introducing GTD points with 10 and 20% of the number required for residual loss calculation is used to predict species concentrations in this ROBER problem. From Figure 2, we can see that the introduction of GTD can improve the accuracy of the prediction result of MPINNs, although the mean squared error (defined by eq 16) of MPINNs with 0% GTD is in the same order as the 10 and 20% cases. In addition, increasing the number size of GTD points from 10 to 20% does not further improve the prediction accuracy, indicating a converging trend with increasing number size of GTD points. In this sense, we shall then use MPINNs with 10% GTD points to further show the superior predictability of MPINNs for the classical chemical kinetic problems in the following sections.

The prediction results of MPINNs with 10% GTD points are then compared to the regular PINN and Stiff-PINN method, using solutions from CVODE as a benchmark, as shown in Figure 3. It is observed that for regular PINNs, the prediction result of y_1 and y_3 has a large error within the time range after 10^0 s, while the prediction result of y_2 shows even greater discrepancy in the entire time range. The Stiff-PINN conjectures that this is due to the stiffness of the ROBER problem.[17] Using QSSA to artificially solve y_2 with algebraic equations, it can be seen that the Stiff-PINN can achieve better predictions.

Without removing the stiffness by artifacts before training, MPINNs proposed in this study first classify species into two groups based on their time scales (see eq 7). Specifically, the species concentrations of $[y_1, y_3]$ and y_2 are obtained from different PINNs with the same structure. The sampling of training points remains the same as the case of regular PINNs. Besides, a small amount of GTD points is added which is 10% of the points we selected for calculating residual loss, where the data are evenly distributed in the sampling results. The residual loss and data sample loss of each species are calculated, and the adaptive weighted summation is carried out to obtain the final loss function for optimization. As shown in Figure 3, the prediction accuracy has been greatly improved compared with the regular PINN method (mean squared error within 10^{-6}, as shown in Figure 2). In addition, it can also be seen from the history of loss functions (Figure 4) that the loss function of MPINNs declines more quickly and is 3 to 4 orders of magnitude lower than that of the regular PINN. As expected, the multiscale structure, the adaptive weight, and the introduction of GTD in MPINNs can make the loss decline rapidly and keep the loss at a relatively low value.

As discussed in the Methodology section, MPINNs proposed in this paper introduce a small amount of GTD

https://doi.org/10.1021/acs.jpca.2c06513
J. Phys. Chem. A 2022, 126, 8534−8543

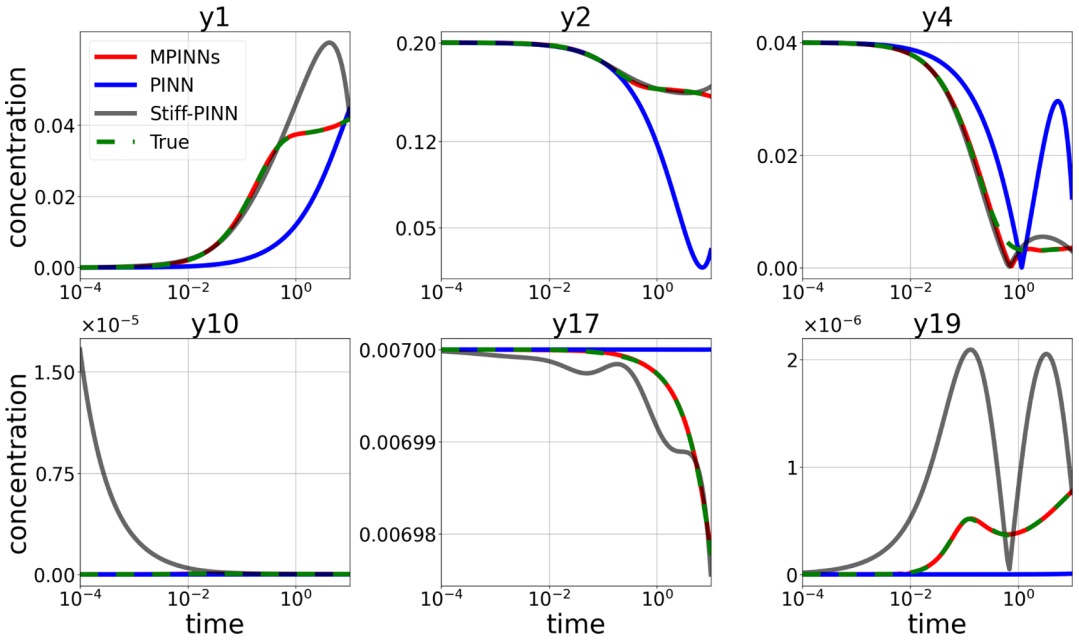


Figure 12. Prediction results of some chemical species concentrations in the POLLU problem obtained by PINNs, Stiff-PINNs, and MPINNs, where the true value (true) is obtained by the traditional CVODE solver.

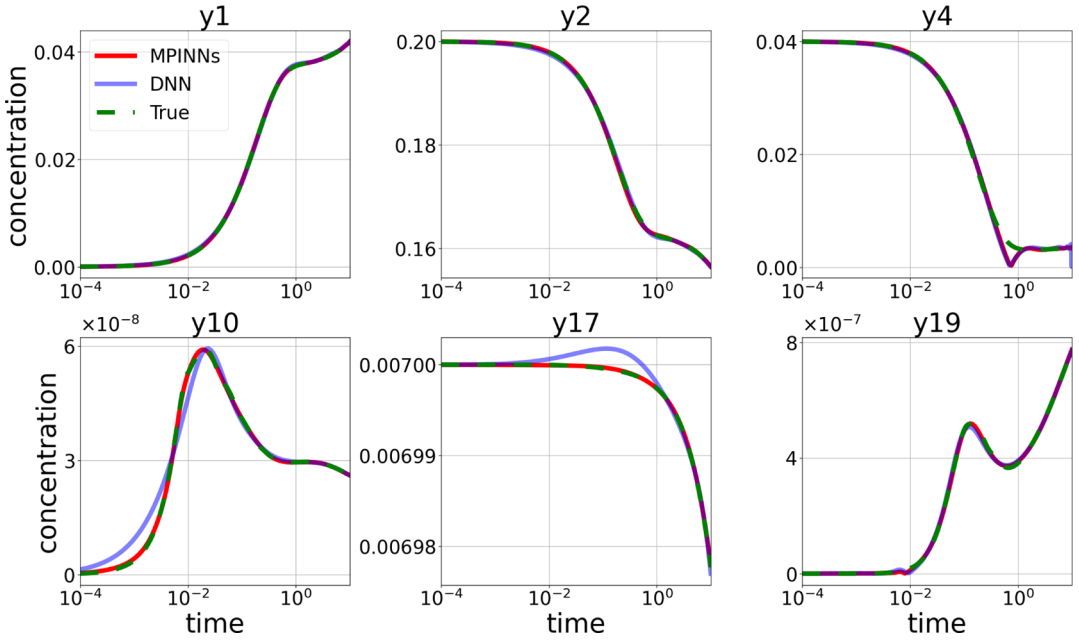


Figure 13. Prediction results of some representative chemical species concentrations in the POLLU problem obtained by MPINNs and DNNs, where the true value (true) is obtained by the traditional CVODE solver.

data into the loss function to calculate the total loss. To show the advantages of this method over traditional DNNs with the same neural network architecture as MPINNs and trained by the same number of data points (i.e., 2500 GTD points), Figure 5 presents the prediction results of DNNs and MPINNs. MPINNs with a small amount (10% of the number required for loss residue calculation) of GTD can achieve similar prediction accuracy on y_1, y_2, and y_3. In fact, Figure 6 shows that MPINNs can even achieve a better accuracy compared with DNNs. As such, MPINNs can be used to solve stiff chemical kinetic problems with a very small amount of data points and achieve even better accuracy than traditional DNNs, eliminating the strict requirement on the size of training data in DNNs.

In addition, the sensitivities of MPINNs to the neural network architecture are briefly discussed here. Different numbers of neurons and layers in the neural networks are tested with MPINNs for the ROBER problem. The performance is found to be not very sensitive to the size (neurons × layers), as shown in Table 1, where the arithmetic average mean squared error is on the same order of magnitude. As such, the architecture with 32 neurons and 3 layers is selected for all the classical chemical kinetic problem solutions by MPINNs in this study.

https://doi.org/10.1021/acs.jpca.2c06513
J. Phys. Chem. A 2022, 126, 8534−8543

Chemical Akzo Nobel Problem. The Chemical Akzo Nobel problem describes a chemical process which originates from the Akzo Nobel Centeral Research in Arnhem, The Netherlands.[24] It includes six chemical species and six reaction equations. The mathematical model describing this set of reactions is shown as

$$\begin{cases} y_1' = -2r_1 + r_2 - r_3 - r_4, \\ y_2' = -\frac{1}{2}r_1 - r_4 - \frac{1}{2}r_5 + F_{in}, \\ y_3' = r_1 - r_2 + r_3, \\ y_4' = -r_2 + r_3 - 2r_4, \\ y_5' = r_2 - r_3 + r_5, \\ y_6' = -r_5, \end{cases}$$

$$\text{s. t.} \begin{cases} y_1(0) = 0.437, \\ y_2(0) = 0.00123, \\ y_3(0) = 0, \\ y_4(0) = 0, \\ y_5(0) = 0, \\ y_6(0) = 0.367 \end{cases} \tag{17}$$

where

$$r_1 = k_1 y_1^4 y_2^{1/2},\ r_2 = k_2 y_3 y_4,\ r_3 = \frac{k_2}{K} y_1 y_5,$$
$$r_4 = k_3 y_1 y_4^2,\ r_5 = k_4 y_6^2 y_2^{1/2} \tag{18}$$

The equations and the values of the parameters can be found in ref 18. Based on the discussions in the previous section, we then fix the neural network architecture (i.e., 32 neurons × 3 layers), 10% GTD points to the data loss term, and use MPINNs to solve the stiff ODEs in this problem. From Figure 7, we can also see that the regular PINN provides very poor prediction for this more complex stiff ODE system compared with the true values solved by the CVODE solver. In MPINNs, we separate all the six chemical species into two groups (as shown in Table 2) based on their time scale orders of the magnitude (eq 7), and each category is fitted with a separate PINN.

As shown in Figure 7, MPINNs offer an accurate prediction for all chemical species in the entire time range. We can also observe the better performance of MPINNs from the loss function history in Figure 8: the loss function of MPINNs stays about 3 order of magnitude lower than that of the regular PINN.

We then compare the prediction accuracy of MPINNs, where only 10% of the number of sampling data points are used with their ground truth values, to that of the conventional DNN with 2500 GTD points (100% of the number of sampling data points). In Figure 9, we can see that MPINNs with a small amount of GTD can achieve qualitatively as accurate prediction as the DNN. Quantitatively, Figure 10 shows that MPINNs possess a better prediction accuracy than that of DNNs for all the six chemical species in the entire time range.

POLLU Problem. We then use MPINNs to solve the more complicated stiff problem—POLLU problem, which includes 20 chemical species and 25 reaction equations. The POLLU problem can be described by 20 different nonlinear ODEs, where details are provided by De Florio et al.[18] In MPINNs, 20 chemical species are first divided into four categories according to their time scale orders of magnitude (see Table 3). The chemical species of each category are fitted with a separate PINN. A small amount of GTD (10% of the number required to calculate the residual loss) is added to the training process. Figure 11 shows the decline of loss function of the regular PINN and MPINN in the training process: similar to the previous problem, it can be observed that the loss function of MPINNs in the POLLU problem decline rapidly and stay at a lower value that that of the regular PINN.

Figure 12 then shows the comparison of six representative species' concentration profiles predicted by the regular PINN, Stiff-PINN, and MPINNs. It can be seen that the prediction error of the regular PINN is significant, failing to predict the dynamics of species change in a wide time span. Although the Stiff-PINN improves the prediction accuracy before $t = 1$, its prediction discrepancy of some chemical species (such as y_1 and y_{19}) is rather obvious. The newly proposed MPINNs offer accurate prediction for all the species profile.

Similarly, we then compared the prediction accuracy of MPINNs to DNNs. In Figure 13, we see that the DNN can achieve reasonably accurate results when using 10 times more GTD points. However, the prediction for some species (e.g., y_{10} and y_{17}) shows obvious discrepancy, indicating that the prediction performance of the traditional DNN for more complicated stiff ODEs relies on data size and data quality. Special data sampling with sufficient data points might be needed as discussed by Zhang et al.[6] to ensure prediction accuracy of the traditional DNN. In contrast, MPINNs can achieve similar or better prediction accuracy when only a few (10% of the amount required for residual loss calculation) GTD samples are used, without special consideration on data sampling treatment.

CONCLUSIONS

In this work, an MPINN method is proposed to solve stiff chemical kinetic problems governed by stiff ODEs. In MPINNs, the chemical species of different time scales are classified in different groups and multiple PINN neural networks are used to predict the residual loss of chemical species in each category. The adaptive weight based on a key performance indicator is then proposed to be added into the process of loss function optimization. During the training process, a small amount of GTD points (10% of the points that are used to calculate the residual loss) is supplemented, and the loss of data is introduced into the loss function during the optimization process. Through the verification on the ROBER problem, Chemical Akzo Nobel problem, and POLLU problem, it is proved that MPINNs are able to precisely learn the solutions for stiff ODEs without stiffness removal artifacts and complex equation transformation. In addition, compared with the DNN optimized by data, this method enables neural networks to greatly reduce the amount of data required through the superposition of residual loss and a small amount of GTD loss. Future studies on MPINNs will focus on stiff ODEs related to different fields or with a variety of initial values.

https://doi.org/10.1021/acs.jpca.2c06513
J. Phys. Chem. A 2022, 126, 8534−8543

AUTHOR INFORMATION

Corresponding Author

Dezhi Zhou − *UM-SJTU Joint Institute, Shanghai Jiao Tong University, Shanghai 200240, PR China;* orcid.org/0000-0001-7808-4767; Email: dezhi.zhou@sjtu.edu.cn

Author

Yuting Weng − *UM-SJTU Joint Institute, Shanghai Jiao Tong University, Shanghai 200240, PR China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpca.2c06513

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

D.Z. gratefully acknowledges the faculty start-up funding from UM-SJTU JI. This work is sponsored by the National Natural Science Foundation of China (grant no. 52106171).

REFERENCES

(1) Ji, W.; Deng, S. Autonomous discovery of unknown reaction pathways from data by chemical reaction neural network. *J. Phys. Chem. A* **2021**, *125*, 1082−1092.

(2) Blasco, J. A.; Fueyo, N.; Larroya, J.; Dopazo, C.; Chen, Y.-J. A single-step time-integrator of a methane-air chemical system using artificial neural networks. *Comput. Chem. Eng.* **1999**, *23*, 1127−1133.

(3) Franke, L. L.; Chatzopoulos, A. K.; Rigopoulos, S. Tabulation of combustion chemistry via Artificial Neural Networks (ANNs): Methodology and application to LES-PDF simulation of Sydney flame L. *Combust. Flame* **2017**, *185*, 245−260.

(4) Ihme, M.; Chung, W. T.; Mishra, A. A. Combustion machine learning: Principles, progress and prospects. *Prog. Energy Combust. Sci.* **2022**, *91*, 101010.

(5) Brown, T. S.; Antil, H.; Löhner, R.; Togashi, F.; Verma, D.Novel dnns for stiff odes with applications to chemically reacting flows. *International Conference on High Performance Computing*, 2021; pp 23−39.

(6) Zhang, T.; Yi, Y.; Xu, Y.; Chen, Z. X.; Zhang, Y.; Weinan, E.; Xu, Z.-Q. J. A multi-scale sampling method for accurate and robust deep neural network to predict combustion chemical kinetics. *Combust. Flame* **2022**, *245*, 112319.

(7) Dikeman, H. E.; Zhang, H.; Yang, S.Stiffness-Reduced Neural ODE Models for Data-Driven Reduced-Order Modeling of Combustion Chemical Kinetics. *AIAA SCITECH 2022 Forum*, 2022; p 0226.

(8) Kim, S.; Ji, W.; Deng, S.; Ma, Y.; Rackauckas, C. Stiff neural ordinary differential equations. *Chaos* **2021**, *31*, 093122.

(9) Donglin, C.; Xiang, G.; Chuanfu, X.; Siqi, W.; Shizhao, C.; Jianbin, F.; Zheng, W. FlowDNN: a physics-informed deep neural network for fast and accurate flow prediction. *Front. Inf. Technol. Electron. Eng.* **2021**, *23*, 207−219.

(10) Shukla, K.; Di Leoni, P. C.; Blackshire, J.; Sparkman, D.; Karniadakis, G. E. Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. *J. Nondestr. Eval.* **2020**, *39*, 61.

(11) Jiang, X.; Wang, D.; Fan, Q.; Zhang, M.; Lu, C.; Lau, A. P. T. Physics-Informed Neural Network for Nonlinear Dynamics in Fiber Optics. *Laser Photon. Rev.* **2022**, *16*, 2100483.

(12) Raissi, M.; Perdikaris, P.; Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686−707.

(13) Li, Y.; Mei, F. Deep learning-based method coupled with small sample learning for solving partial differential equations. *Multimed. Tool. Appl.* **2021**, *80*, 17391−17413.

(14) Chakraborty, S. Transfer learning based multi-fidelity physics informed deep neural network. *J. Comput. Phys.* **2021**, *426*, 109942.

(15) Wang, S.; Teng, Y.; Perdikaris, P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.* **2021**, *43*, A3055−A3081.

(16) Xiang, Z.; Peng, W.; Liu, X.; Yao, W. Self-adaptive loss balanced Physics-informed neural networks. *Neurocomputing* **2022**, *496*, 11−34.

(17) Ji, W.; Qiu, W.; Shi, Z.; Pan, S.; Deng, S. Stiff-pinn: Physics-informed neural network for stiff chemical kinetics. *J. Phys. Chem. A* **2021**, *125*, 8098−8106.

(18) De Florio, M.; Schiassi, E.; Furfaro, R. Physics-informed neural networks and functional interpolation for stiff chemical kinetics. *Chaos* **2022**, *32*, 063107.

(19) Guo, M.; Haque, A.; Huang, D.-A.; Yeung, S.; Fei-Fei, L.Dynamic task prioritization for multitask learning. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018; pp 270−287.

(20) Liu, S.; Johns, E.; Davison, A. J.End-to-end multi-task learning with attention. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019; pp 1871−1880.

(21) Cohen, S. D.; Hindmarsh, A. C.; Dubois, P. F. CVODE, a stiff/nonstiff ODE solver in C. *Comput. Phys.* **1996**, *10*, 138−143.

(22) Hendrycks, D.; Gimpel, K.Gaussian error linear units (gelus). **2016**, arXiv:1606.08415. arXiv preprint, submitted on 27 Jun, 2016,

(23) Kingma, D. P.; Ba, J.Adam: A method for stochastic optimization. **2014**, arXiv:1412.6980. arXiv preprint.

(24) Eriksson, K.; Johnson, C.; Logg, A. Explicit time-stepping for stiff ODEs. *SIAM J. Sci. Comput.* **2004**, *25*, 1142−1157.